Multi-Layered Friendship Modeling for Location-Based Mobile Social Networks

Nan Li and Guanling Chen
Department of Computer Science, University of Massachusetts Lowell
1 University Avenue, Lowell, MA 01854
{nli, glchen}@cs.uml.edu

***Abstract*—Location-based Mobile Social Networks (MSNs) are becoming increasingly popular given the success of Online Social Networks (OSNs), such as Facebook and MySpace, and recent availability of open mobile platforms, such as Apple iPhones and Google Android phones. MSNs extend existing OSNs by allowing a user to know when her friends are around and by providing the ability to meet new people who share her interests. There are few studies, however, on how users are connected through these emerging location-based MSNs.**

In this paper, we present analysis results of a commercial MSN for which we quantified the correlation between users' friendship with their mobility characteristics, social graph properties, and user profiles. The evaluation of the derived model from the empirical traces suggests that the model-based friend recommendation is effective, and its performance is better than well-known Naive Bayes classifier and J48 decision tree algorithms. To the best of our knowledge, this paper presents the first study that models the friendship connections over a real-world location-based MSN.

I. Introduction

In recent years, online social networks (OSNs), such as Facebook and MySpace, have become extremely popular with more than half billion worldwide users. With the increasing availability of smartphones and open mobile platforms, users can access those services from anywhere at anytime. A recent study shows that the top Web destinations on smart mobile phones are OSN sites [23] and there are already many social network applications available on popular mobile platforms [9], [3].

OSN services allow users to easily share thoughts, activities, photos, and other information with friends to strengthen their connections. This kind of sharing of user-generated content, sometimes called status updates or micro-blogging [5], is becoming extremely popular with successful services like Twitter.[1] Mobile Social Networks (MSNs) can further facilitate users to share their status updates anytime and anywhere. In particular, it is also natural for MSN applications to share a user' current location, which can be broadcast to her friends or be used to tag her updates. Many recent smartphones support GPS-based or signal-triangulation based localization, making it easier for MSN applications to access and use users' location. It is estimated that location-based social networking will generate \$3.3 billion revenue by 2013 [18].

Typically there are two types of usage patterns for social networks. One is to allow users to strengthen the connections with their existing friends, such as sharing "what are they doing" through status updates. For the location-based MSNs, users may also share "where are they right now" with the mobile devices' localization capability. A case study of Dodgeball[2] shows how the MSN changes users' communication patterns between friends [10]. The other purpose of using the social networks is for the users to make new friends, such as based on common interests. Location-based MSNs make it possible to select potential new friends based on geographic proximity, so it is more likely that they can meet in person. For example, both Loopt[3] and Brightkite[4] allow users to "snoop" nearby non-friends' public status updates, to discover those in the region who may share some common interests.

In practice, however, there are few studies on how users are connected through these emerging social networks. Intuitively, we understand that the formation of online friendship may reflect to some degree of the real-world counterpart. Namely, people who have mutual interests, are geographically close, or belong to common social circles (friends of friends), are more likely to become friends. The challenge is what metrics can be used to represent these intuitive understandings with a quantitative model. While there are quite a few studies on social networks' structural properties, such as node degree distributions, we have found little work on modeling friendship (connections) with other user attributes.

If we have a quantitative friendship model, we can build an automated tool that recommends possible new people for a user to connect with. This is particularly useful since a social network can be quite large with many status updates, thus prohibiting manual inspections. For example, Figure 1 shows the number of daily active users and their updates for a location-based MSN, Brightkite, during a 23-day period. The average number of daily active users and their updates were as large as 3,656 and 9,945, respectively. So with a model-based friendship recommendation algorithm, it is much easier for mobile users to find new people to meet with, thus encouraging more real-world interactions through the use of MSNs.

In this paper, we present a friendship model for a location-based MSN (Brightkite), in which each update is attached with user's location. We selected 18,951 active Brightkite users

[1]http://www.twitter.com/

[2]http://www.dodgeball.com/

[3]http://www.loopt.com/

[4]http://www.brightkite.com.

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6828
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6828

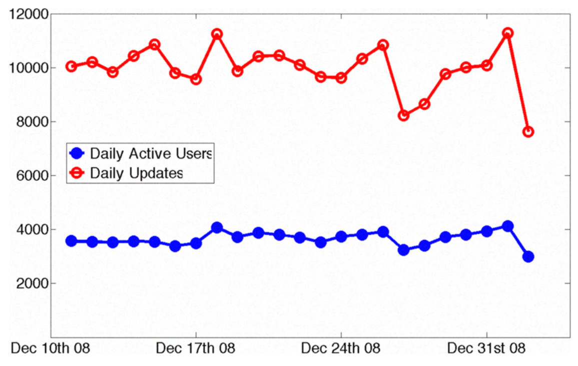

Fig. 1. The number of daily active users and their updates for Brightkite.

and collected all their 1,505,874 updates as the training data set. In addition, we also collected users' profiles and their friend lists. We then built a three-layered friendship model to correlate the relationship between users' friend connections with attributes of their profiles, social graphs, and mobility patterns. To validate the proposed model, we used new friend connections during the 45 days after the time of training data. We found that the proposed model can better predict user friends than well-known Naive Bayes and J48 Tree algorithms, thus providing a valuable foundation for friend recommendations in MSNs. To the best of our knowledge, this paper presents the first study that models the friendship connections over a real-world location-based MSN.

The rest of this paper is organized as follows. In Section II, we discuss the related work. Section III presents some background for location-based MSNs. Section IV describes how we collected the data traces and Section V presents the three-layered modeling. The evaluation results are shown in Section VI, and Section VII provides further discussions. Finally, we conclude in Section VIII.

II. Related Work

There have been quite a few measurement studies in OSNs, most of which have focused on structural analysis of the social graphs [4], [19], [11]. Other researchers have also studied the changing dynamics of complex networks. For example, Barabási et al. studied preferential attachment model, where the likelihood of receiving new edges (friendships) increases with node's degree [1], which is validated in real large-scale OSNs [13]. These studies, however, are statistical in nature and are not suitable for individual friend recommendations.

Schwartz and Wood propose an "interest distance" metric measuring the similarity of two users' friend circles, based on which a clustering algorithm is used to derive a list of users whose interest is close to the target user [22]. The evaluation of this approach using a social graph derived from Email communications shows noisy results that can be improved using domain-specific heuristics. Grob et al. use the CONGA community detection algorithm [6], to make iterative friend recommendations from the same community [7]. These approaches, however, leverage only the social graph information and make an assumption of correlation between the shared interest and the graph structures.

Another line of research related to our work is the *Recommendation System*, which typically uses a specific type of information filtering technique to suggest items (movies, books, etc.) that are likely of interest to the user. The e-commerce Web sites, such as Amazon.com, are among the early adopters of these recommendation systems [16], where a list of recommended items are generated using several user attributes, including items viewed, demographic data, subject interests, and so on.

The *Collaborative Filtering* (CF) [21], is a popular technique used by the recommendation systems. CF makes predictions about a user's preferences based on the preferences of a group of users that are considered similar to this user. There are generally two approaches to determine user similarity. Content-based methods use similarity metrics to express a distance between two users [8]. Model-based methods use the ratio metric, that contains the ratings of all users who have expressed their preferences, to create a model from which the sets of similar users will be established [2].

Different than these recommendation systems, this paper focuses on user to user friendship modeling instead of connections between users and items. We leverage the attribute divergences between friend pairs and non-friend pairs to the classification model.

In industry, there are a few Web sites addressing the friends suggestion problem. For example, The Tweetsum,[5] Mr. Tweet,[6] and Twitseeker[7] focus on recommending friends for micro-blogging service Twitter. Twitter itself also provides user suggestions.[8] Most of these tools suggest friends by analyzing the users' update content or their popularity, though no details of their analysis algorithms are available in public. Our study focuses on modeling friendship over location-based MSNs, and the model can be used to recommend people a user is more likely to meet in person.

III. Location-based Mobile Social Network

Dodgeball[9] is one of the first location-based mobile social network applications that allow a user to broadcast her current location to selected friends or even to an extended list of friends-of-friends whose mobile phones are, for instance, within a 10-block radius [10]. Dodgeball relies heavily on SMS to allow users with virtually any mobile phone to connect and share while on the move. On the other hand, Loopt[10] leverages GPS and signal triangulation technologies to automatically sense device location, without requiring manual location updates. The downside of Loopt, however, is that it is restricted to compatible phones and participating carriers.

[5]http://tweetsum.com/
[6]http://mrtweet.net/
[7]http://twitseeker.com/
[8]http://twitter.com/invitations/suggestions
[9]http://www.dodgeball.com/
[10]http://www.loopt.com/

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6828
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6828

Brightkite is a recent Denver-based startup, founded in 2005, that allows users to share their location, to post notes, and to upload photos through a number of interfaces, including Web, SMS, and Email. It is, however, also possible for users to update Brightkite using Web or Email applications, instead of SMS, on their smartphones. Recently, the company has also released a native client application on Apple iPhone and is planning a version for Google Android phones. These native client applications, like Loopt clients, leverage GPS and other on-device technologies for automatic location sensing, though still requiring users to hit "check in" button to update their current location.

Brightkite allows users to define their friends and subscribe to their *activity streams*, including locations they checked in, their posted notes, and their uploaded photos. A note is limited to contain maximum of 140 characters, for users to share quick thoughts and short status updates. The "friendship" relation is mutual: a user X accepting Y's friend request means that X and Y become each other's friend. A user may choose to *protect* her activity stream so only her friends can see her location/note/photo updates. A user may discover nearby people and browse their activity streams, if they are not protected.

The posted notes and the photos are all attached with user's most recent checked-in location. Once a user checked in at a location, she is assumed to stay at that place until she explicitly checked in at another location. This mechanism gives users a complete control on when and where to share their location, addressing some privacy issues when sharing sensitive location information. On the other hand, users may not always remember to update their location as they move around, leading to incorrect location perceived by their friends. It is debatable and unclear in reality, whether users prefer this manual location updates to automatic location sensing adopted by Loopt. When sharing user's current location, Brightkite allows users to control the "granularity". Namely, users can check in at a country, a city, a zip code or the exact address.

IV. Data Collection

Brightkite website provides a public-accessible RESTful API for integration with the third-party applications. The interface is implemented by HTTP query, so an application sends a specified HTTP request and the website sends back an XML response. An example XML-represented user note update is shown in Figure 2.

To collect users' updates, we first queried all public updates that were posted between December 9, 2008 to January 9, 2009. There are 18,951 users who made at least one update during that period. Our analysis thus focused on these users, because we believe that the other users are inactive since they did not make any update for a month. After determining the user population, we then queried all updates for each user from March 21, 2008 (the earliest date we could get a user update from their website) to January 9, 2009. Brightkite returned 1,505,874 updates in total for these 18,951 users. We also

```
<note>
 <created_at_ts>1234822696.004165
</created_at_ts>
 <body>in the umpteenth surreal moment between
  nice memories and a sweet doubt. the
requirements are good to have a nice sleep...
</body>
 <creator>
  <fullname>Gennaro Del Giudice</fullname>
  <login>gdelgiudice</login>
 </creator>
 <public type="boolean">true</public>
 <place>
  <scope>address</scope>
  <display_location>Viale Fulvio Testi, 280,
   20126 Milano, Italy</display_location>
  <longitude type="float">9.21059</longitude>
  <name>Viale Fulvio Testi, 280</name>
  <id>2d3335eedb6b11ddbf3f003048c10834</id>
  <latitude type="float">45.523338</latitude>
 </place>
 <created_at type="datetime">
2009-02-16T22:18:16Z</created_at>
 <id>b5162494fc7711ddb34c003048c10834</id>
</note>
```

Fig. 2. An example XML-represented user note update.

collected all these users' profile information, including age, gender, tag list, and friend list.

In order to validate the friendship model, we need to know how the users' friend lists change over time. We first queried a "full" social graph, which included 41,014 users in total as of January 9, 2009. To get the full social graph, we queried the friend lists in multiple rounds. The querying started from the 18,951 users and retrieved each user's friend list. Then we went through those friend lists, recording the unvisited users for the next round querying. We kept repeating the process until no new user was found. This way we got a snapshot of the social graph seeded by the 18,951 users. Then we queried the daily active users' friend lists everyday from January 10 to February 25, 2009. By comparing the daily friend lists with the previous social graph snapshot, we were able to capture the daily changes of the social graph for 45 days. There were 5,098 new friend pairs created during that period.

Table I shows some statistics of the activity updates and the active users. About 48.8% of public activities were check-in updates, when users explicitly updated their current locations. Another 38.6% of updates were text notes and 12.6% were photos shared by users. About 50.0% updates were attached with location of "address" scope, which provided up to 0.1 mile location accuracy. About 31.5% updates were "city" scope, which was a much less accurate level. We assume that city level accuracy is between 10 to 30 miles range.

Users may tag their profiles with keywords that describe their jobs or interests. By aggregating these tags, we may have a basic understanding what kinds of people use Brightkite. Figure 3 shows the cloud of top 50 tags, with larger font size indicating more frequent appearance in users' profiles. It is

Total Updates		1,505,874	
types	check-in	734,995	48.8%
	note	581,030	38.6%
	photo	189,849	12.6%
scopes	address	753,644	50.0%
	city	474,662	31.5%
	zip	89,177	5.9%
	street	67,620	4.5%
	zip+4	52,782	3.5%
	others	67,989	4.5%
Total Users		18,951	
gender	male	12,686	66.9%
	female	2,525	13.3%
	unspec	3,740	19.7%
age	≤ 20	383	2.0%
	$20 \sim 30$	4,383	23.1%
	$30 \sim 40$	3,099	16.4%
	≥ 40	1,152	6.1%
	unspec	9,934	52.4%

TABLE I
TRACE DATA SUMMARY

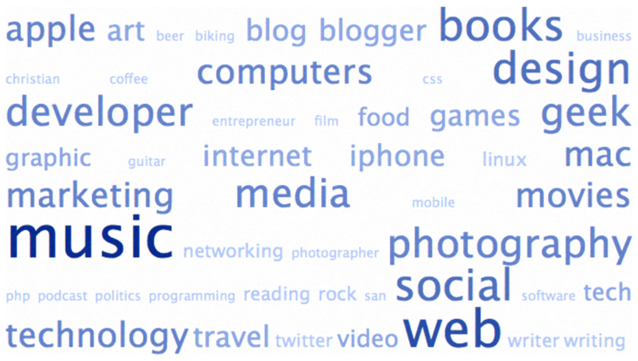

Fig. 3. Cloud of top 50 tags in users' profiles (the font size indicates the frequency).

not surprising to see that many users seem to be interested in marketing over social media, which is primarily Internet and mobile-based tools, such as Brightkite, for sharing and discussing information. Other popular tags include *music*, *books*, and *photography*, which reflect personal interests.

V. Multi-Layered Friendship Modeling

In social communities, friend connections have strong correlations with users' social graph properties [1], [13]. In our analysis, we exploit the relationship between the users' social graph distance with their friendship. It is also commonly accepted that friends are likely to share some common interests [17], reflected to some degree by Brightkite users' profile tags. Thus we also quantify the relationship between users' tag distance with their friendship. In addition, we found that with high probability friends will send updates within small distance range observed over location-based MSNs, such as Brightkite. Based on those considerations, we propose a three-layered model to exploit the correlations of user's

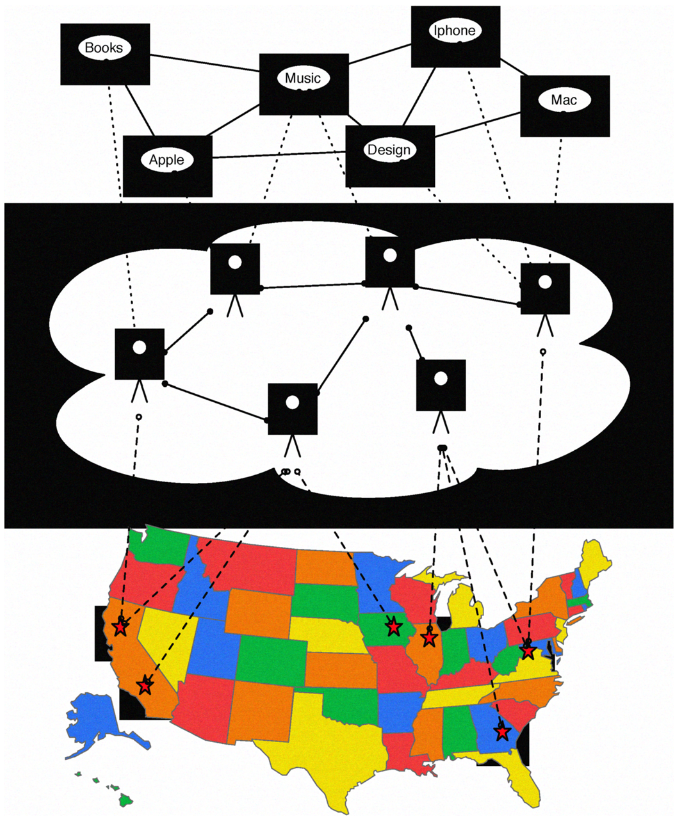

Fig. 4. Three-layered friendship modeling for location-based MSNs.

friendship with social graph properties, tag attributes, and mobility patterns. Figure 4 shows an illustrative diagram of the three-layered friendship model.

In a social network $G_s(V_s, E_s)$ where each node v_s denotes a user and each edge e_s denotes the friendship between two nodes. For any node (user) pair (a, b), we can calculate the *pair's* attribute(s). Based on the attribute values, we can compute the empirical probability of being friends. Assuming that m_v is the attribute value, $P_f(m_v)$ denotes the probability of value m_v for friend pairs and $P_{nf}(m_v)$ denotes the probability of value m_v for non-friend pairs. Using the *Bayes' rule* of conditional probability, the probability $P(ab|m_v)$ of existence of edge between the two users (meaning they are friends) while observing attribute value m_v is,

$$P(ab|m_v) = \frac{P(m_v|ab) \times P(ab)}{P(m_v)} \qquad (1)$$

$P(m_v|ab)$ denotes the conditional probability of given (a, b) are friends and the m_v occurrence. $P(ab)$ denotes the probability of node a and b are friends. $P(m_v)$ denotes in all possible pairs the probability of attribute value m_v occurrence. We have

$$P(m_v|ab) = P_f(m_v) \qquad (2)$$

$$P(ab) = \frac{|E_s|}{\mathbf{C}^2_{|V_s|}} \qquad (3)$$

$$P(m_v) \cong P_{nf}(m_v) \qquad (4)$$

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6828
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6828

The $\mathbf{C}^2_{|V_s|}$ denotes the all possible user pairs. Equation 4 is satisfied because $\mathbf{C}^2_{|V_s|}$ is much larger than $|E_s|$. In current social graph, the $|E_s| = 46,172$ and $\mathbf{C}^2_{|V_s|} = 18,951 \times 18,950/2 \cong 1.8 \times 10^8$. The probability of m_v in non-friend pairs is approximate to the one in all possible pairs. Considering the Equations 2, 3 and 4, the Equation 1 becomes

$$\begin{aligned} P(ab|m_v) &= \frac{P_f(m_v) \times |E_s|}{P_{nf}(m_v) \times \mathbf{C}^2_{|V_s|}} \\ &= F(m_v) \times \frac{|E_s|}{\mathbf{C}^2_{|V_s|}} \end{aligned} \quad (5)$$

We define the *rank factor* $F(m_v)$ as,

$$F(m_v) = \frac{P_f(m_v)}{P_{nf}(m_v)} \quad (6)$$

Since the value of $|E_s|/\mathbf{C}^2_{|V_s|}$ is constant for a different user pairs, we need only calculate $F(m_v)$ and compare its value over different user pairs to determine which pair has higher probability to be friends.

We have the existing 46,172 friend pairs identified by January 9, 2009. The $P_f(m_v)$ can be calculated using this empirical data. Since the number of non-friend pairs is huge, it is hard to explore all sample space. Thus we randomly selected 100,000 non-friend pairs and calculated $P_{nf}(m_v)$ from those samples.

In reality, a single attribute is not always enough for friend recommendation. For example, potential friends could share same tag(s) because of mutual interesting(s), but may never send updates in close-by locations. Thus if we use the location-based attribute only, we may not accurately model the pair relationship. We thus selected three attributes (details later) to calculate the probability of friendship separately. Assuming that P_1, P_2, and P_3 denote the probabilities calculated using the three different attributes, and if they are independent, we have,

$$\begin{aligned} P(ab|m_{v1,v2,v3}) &= 1-(1-P_1)\cdot(1-P_2)\cdot(1-P_3) \\ &= 1-(1-P_1-P_2+P_1\cdot P_2)\cdot(1-P_3) \\ &= P_1+P_2+P_3+O(P^2)-O(P^3) \\ &\cong P_1+P_2+P_3 \end{aligned} \quad (7)$$

We can ignore the $O(P^2)$ and $O(P^3)$, because the probability is extremely small. Considering the Equation 5, in our current social graph, $|E_s|/\mathbf{C}^2_{|V_s|} = 2.5 \times 10^{-4}$, and each rank factor $F(m_v)$ is usually less than 300.

By ignoring the constant value, we can rank users' friendship by comparing the summarization of the three attributes' rank factors.

$$F = F_1 + F_2 + F_3 \quad (8)$$

In Equation 8, if one or two attributes are missing, the model is still valid. Next we discuss the three attributes used in our model.

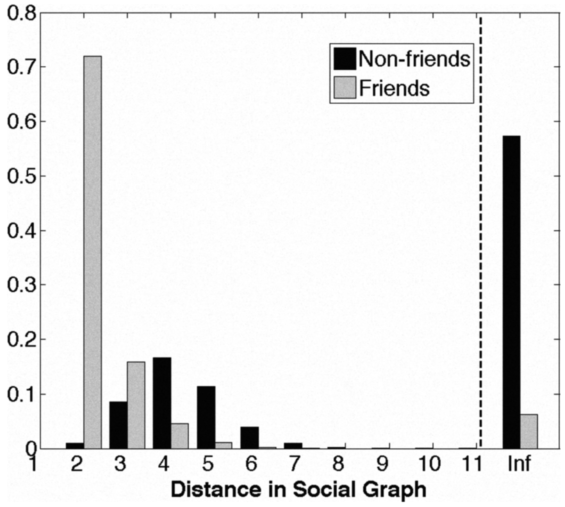

Fig. 5. Histogram of Distance DS of friends/non-friends in the social graph. The infinite value means that the node pair is disconnected.

A. Social Graph Layer

In Figure 4, the middle layer is the social graph which is denoted by $G_s(V_s, E_s)$. In the graph G_s, each node v_s is a Brightkite user, and we have $|V_s| = 18,951$. Each edge e_s denotes a pair of friends, and we have $|E_s| = 46,172$. Since the "friendship" in Brightkite is mutual, G_s is an undirected graph. We define the *friendship distance* $DS_{i,j}$ of v_i and v_j in Equation 9, $e_{i,j}$ is the link between v_i and v_j.

$$DS_{i,j} = shortest\_distance(v_i, v_j) \text{ in } G'_s(V_s, E_s - e_{i,j}) \quad (9)$$

The Equation 9 means if v_i and v_j are friends, we first remove the existing edge between i and j, then calculate their shortest path in the remaining social graph.

Figure 5 shows the histogram of normalized DS including non-friend pairs (left bar) and friend pairs (right bar). We found that more than 70% friend pairs' distance is 2 and more than 15% friend pairs' distance is 3. On the other hand, very few non-friend pairs' (less than 9%) distance is less than 4. The majority of non-friend pairs (58%) are disconnected nodes in the social graph, which suggests that the social graph in Brightkite is quite disconnected.

We calculated the ranking factor of social graph layer by dividing the friend pairs' proportion by non-friend pairs' proportion (Equation 6). We consider the situations that the rank factor larger than 1 and ignore the other situations. Then we have,

$$F_s(m_s) = \begin{cases} 66.86 & m_s = 2 \\ 1.46 & m_s = 3 \\ 0 & others \end{cases} \quad (10)$$

B. Tag Graph Layer

To model the relationship between friendship and users' tags, we selected 1000 most popular tags to build the tag graph, the upper layer graph in Figure 4, denoted by $G_t(V_t, E_t)$. Since the Brighkite allows users to use any words as their tags, we only focused on the popular tags to get the statistics. G_t is an undirected **complete** graph. We connect each pair of tags with a non-negative value w_t as weight to indicate the tag closeness, which is defined as,

$$\begin{aligned} w_{t_1,t_2} &= \frac{P_f(t_1,t_2)}{P_a(t_1,t_2)} \\ &= \frac{P_f(t_1,t_2)}{P_a(t_1)\cdot P_a(t_2) + P_a(t_2)\cdot P_a(t_1)} \\ &= \frac{P_f(t_1,t_2)}{2\cdot P_a(t_1)\cdot P_a(t_2)} \end{aligned} \tag{11}$$

$P_f(t_1,t_2)$ denotes the probability of tag t_1 and tag t_2 shared by friends in all friend pairs. We calculated this by counting the number of pairs, where one user had tag t_1 and the other had tag t_2, in friends pairs to get $CNT_f(t_1,t_2)$. We then divided it by the total number of friends pairs $|E_s|$. Note it is possible that t_1 and t_2 are the same.

$$P_f(t_1,t_2) = \frac{CNT_f(t_1,t_2)}{|E_s|} \tag{12}$$

$P_a(t_1,t_2)$ denotes the probability of for one user pair, one user had tag t_1 and the other had tag t_2, in all possible user pairs, and $P_a(t_1)$ denotes the probability of tag t_1 occurring in all users. We calculated the $P_a(t_1)$ by counting the number of users who used tag t_1 in all users to get $CNT_a(t_1)$ and divided it by the total number of users $|V_s|$.

$$P_a(t_1) = \frac{CNT_a(t_1)}{|V_s|} \tag{13}$$

Since we are interested in the ranking of tags' closeness, we eliminate the constant, and denote the w_t as,

$$w_{t_1,t_2} = \frac{CNT_f(t_1,t_2)}{CNT_a(t_1)\cdot CNT_a(t_2)} \tag{14}$$

if either t_1 or t_2 is not in the 1000 top tag list, we assign the weight $w_{t_1,t_2} = 0$. The high weight tag pairs include (*music*, *photography*), (*photography*, *travel*), and so on. The low weight tag pairs include (*film*, *php*), (*linux*, *snowboarding*), and so on.

Assuming that the user i's tag list is $(t_{i1}, t_{i2}, \cdots, t_{iM})$ and the user j's tag list is $(t_{j1}, t_{j2}, \cdots, t_{jN})$, we define the tag metric $m_{t(i,j)}$ as,

$$m_{t(i,j)} = \sum_{n=1}^{N}\sum_{m=1}^{M} w_{t_{im},t_{jn}} \tag{15}$$

Figure 6 shows the histogram of tag metrics including both friend pairs (right bar) and non-friend pairs (left bar). The friend pairs' tag metric distribution is larger than non-friend pairs for most metric values, especially in large values range, which means friend pairs tend to share closer tags. We then calculated the rank factor of tag metric as,

$$F_t = \frac{P_f(m_t)}{P_{nf}(m_t)} \tag{16}$$

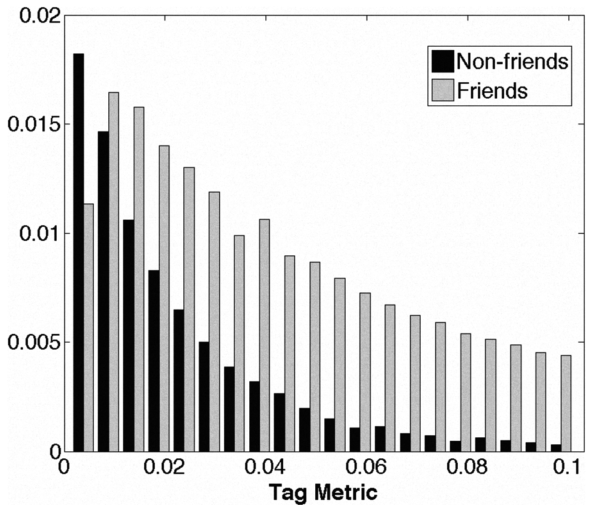

Fig. 6. Histogram of tag metric M_t of friends and non-friends.

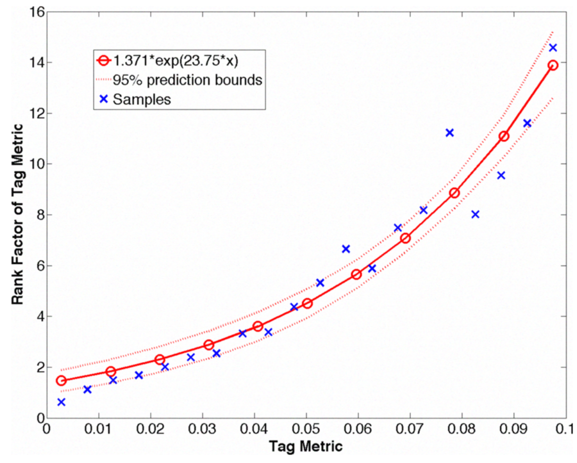

Fig. 7. Rank factor of the tag metric.

In order to guarantee the statistical significance of $P_{nf}(m_t)$, we required that the tag count was at least 50 for each sample. So we ignored the value of $P_{nf}(m_t)$ when $m_t > 0.1$. Figure 7 shows the samples of F_m by cross markers. We applied exponential function to fit the samples and got a fitting function, shown as the point-line. The two dot lines show the 95% prediction bounds by the fitting function. We noticed that the samples go to infinite when the tag metric is larger than 0.1, and we arbitrarily assigned a big number, 100, in those situations.

The F_t can be denoted as,

$$F_t(m_t) = \begin{cases} 0 & m_t = 0 \\ 1.371 \times e^{(23.75\times m_t)} & 0 < m_t \le 0.1 \\ 100 & 0.1 < m_t \end{cases} \tag{17}$$

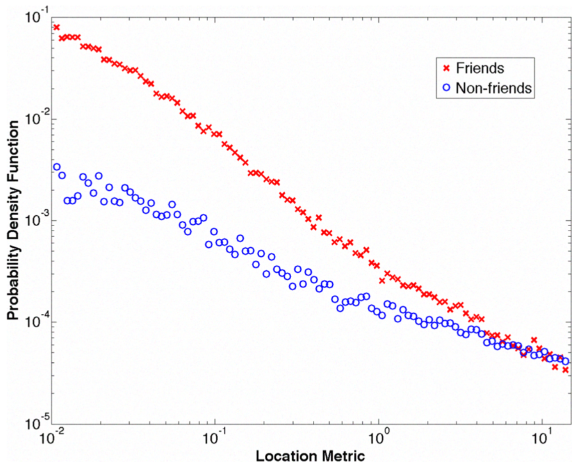

Fig. 8. Probability of the location metric.

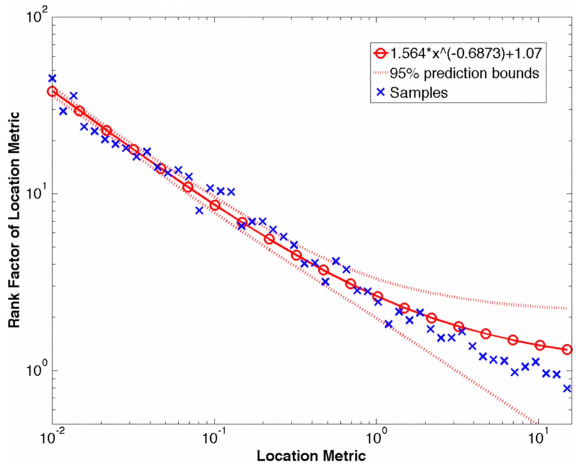

Fig. 9. Rank factor of the location metric.

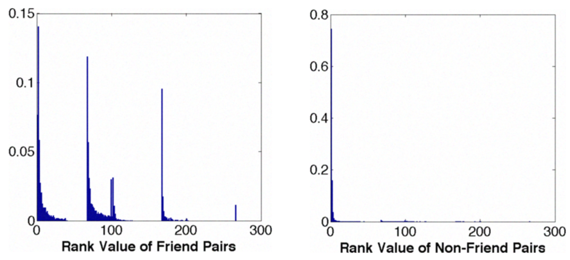

Fig. 10. Histogram of the ranking values.

C. Location Graph Layer

In Figure 4, the lower layer denotes the location based graph $G_l(V_l, E_l)$ of user updates. Although the figure shows a U.S.A map, the actual user updates in our data distribute over the global area.

Assuming that the user i sent updates in the locations $(l_{i1}, l_{i2}, \cdots, l_{iN})$, and each location contained the number of updates $(c_{i1}, c_{i2}, \cdots, c_{iN})$, respectively. $Dist(l_1, l_2)$ denotes the geographical distance of locations l_1 and l_2.

We defined the location metric $m_{l(i,j)}$ between user i and j as,

$$m_l(i,j) = \min_{\substack{\forall m \in (1,M), \forall n \in (1,N) \\ \text{when}, \quad Dist(l_{im}, l_{jn}) < 30}} \left(\frac{Dist(l_{im}, l_{jn})}{c_{im} + c_{jn}} \right) \tag{18}$$

In Equation 18, we ignored the location pairs whose distance is larger than 30 miles. We believe that the geographical correlation of two locations that are 30 miles away can be ignored. The location metric is the minimum value of updates distance divided by sum of update times in the two locations. We assume that user pair (A, B) are closer than user pair (C, D), if (A, B) have more updates than (C, D), even if their update distances are same. More updates (A, B) means that they could be neighbors or work in nearby buildings. On the other hand, few update times (C, D) means that they may happen to visit the same place and are not so close as (A, B).

Figure 8 shows the location metric's probability density function for both friend pairs and non-friend pairs. Friend pairs tend to have smaller values of the location metric. Similar to tag metric, we define the rank factor of location metric as,

$$F_l = \frac{P_f(m_l)}{P_{nf}(m_l)} \tag{19}$$

Figure 9 shows the samples of F_l by cross markers. We applied power law function to fit the samples and got a fitting function, shown as the point-line. The two dot lines shows the 95% prediction bounds by the fitting function. We ignored the situations where $m_l > 15$ because the $F_l < 1$, and arbitrarily assigned a big number, 100, when $m_l < 0.01$.

$$F_l(m_l) = \begin{cases} 100 & m_l < 0.01 \\ 1.564 \times m_l^{-0.6873} + 1.07 & 0.01 \le m_l \le 15 \\ 0 & 15 < m_l \end{cases} \tag{20}$$

D. Friendship Ranking

As we describe at the beginning of Section V, given any user pair, we can calculate the ranking value as,

$$F(m_s, m_t, m_l) = F_s(m_s) + F_t(m_t) + F_l(m_l) \tag{21}$$

The larger value $F(m_s, m_t, m_l)$ is, the higher probability for two users to be friends. For any given user, we can calculate the ranking value with other users and provide a high-ranking user list to her as the recommended friends.

VI. Evaluation

To validate the proposed friendship model, we collected the changes of users' friend lists for 45 days and found 5,098 new friend pairs. We also randomly selected another 100,000 non-friend pairs which have no overlap with the 100,000 non-friend pairs used for model training. The two groups of user pairs were test sets for model evaluation.

Figure 10 shows the ranking values' distributions of the two test sets. It is quite obvious that the distributions of

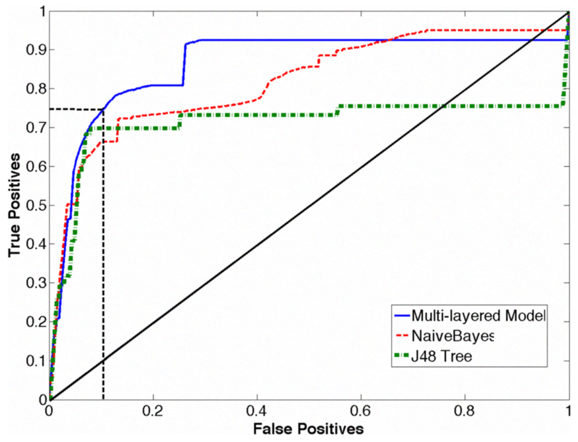

Fig. 11. ROC curves of the friendship ranking.

friend pairs and non-friend pairs are different. For example, there are 78.42% friend pairs whose ranking value is larger than 2.5, by adding all the bars with $x > 2.5$, while there are 90.16% non-friend pairs whose ranking value is smaller than 2.5. The spikes in the first figure indicate the common ranking values with the large numbers 66.86, 100, and 100 in Equations 10, 17, and 20, respectively.

A. Model Prediction Results

We used the ROC curve to compare the ranking performance of the proposed model and other modeling algorithms, as shown in Figure 11.

To plot the ROC curve, we first calculated the ranking values of all test sets (100,000+5,098 user pairs), and sorted them in decreasing order. We went through the sorted ranking values from top and drew points in ROC figure from original point (0,0). If the current ranked user pair was indeed a friendship, we moved the cursor along y-axis with normalized distance $dy = 1/5098$ and drew a point. If the current ranked user pair was not a friendship, we moved along x-axis with distance $dx = 1/100000$ and drew a point. After drawing a point, we check the next ranked value and repeat the process again until the end of data set. If we consider the ranking result as friendship prediction, the x-axis denotes the false positives and y-axis denotes the true positives.

For perfect friendship ranking, all the true friend pairs should aggregate at the top rank values, the ROC curve thus yields the line from original point (0,0) to upper left corner (0,1) and to upper right corner (1,1). A completely random ranking, however, would give a diagonal line from the left bottom to the top right corner.

The non-diagonal solid line is the ROC curve for the proposed model. We found that more than 70% true friend pairs against about 10% false positives. In other words, by using the friendship ranking, about 70% friend pairs aggregate at the top 10% data set.

To compare the modeling performance with other data mining algorithms, we selected one classifier algorithm, Standard Probabilistic NaiveBayes [12], and one decision tree algorithm, C4.5 revision 8 decision tree learner (J48) [20], and applied the same data sets as used by our model and evaluated their performance results. The Figure 11 also shows these two algorithms' performance, the dashed line denotes NaiveBayes algorithm and the dotted line denotes the J48 decision tree algorithm. The results suggest that our model show the best performance and the NaiveBayes was better than J48 Tree.

To quantify the overall performance, we calculated the ROC Area, which is the area below the ROC curve. Of course, the ROC Area of perfect ranking is 1 and random ranking is 0.5. Table II shows the three algorithms' comparison result, the multi-layered model has larger ROC Area than the other two models, indicating better performance.

Multi-layered	NaiveBayes	J48 Tree
0.865	0.845	0.785

TABLE II
THE ROC AREA OF THREE ALGORITHMS

B. Location Metric

One of this paper's goals is to exploit the relationship between location-based metric with users' friendship for MSNs. We calculated the ROC area of the proposed model without location-based metric. In this case, the model has two layers, social graph and tag graph. The ROC area is 0.761, with significant performance reduction compared with the three-layer model.

Additionally, we used the *information gain* to quantify the impact of different attributes to friendship prediction. Besides the existing three attributes that we have discussed, we also calculated the information gain of users' gender difference and age difference in friendship prediction. Table III shows the results.

Attributes	Info. Gain
Social Dist.	0.574
Loc. Metric	0.345
Tag Metric	0.078
Gender Diff	0.030
Age Diff	0.012

TABLE III
INFORMATION GAIN OF DIFFERENT ATTRIBUTES

It shows that the location-based metric provided comparable information gain as the social graph distance in friendship prediction. The tag metric has smaller information gain than the other two metrics, which is likely because many users used non-popular tags (out of the top 1000), and their tag metrics become zero. The users' gender and age attributes, however, provide little information gain, thus do not help predict friendship and our model does not include them. For comparison, Leskovec and Horvitz studied the social graph formed by instant messaging (Microsoft Messenger in this case), and also found that location attribute is highly correlated with user

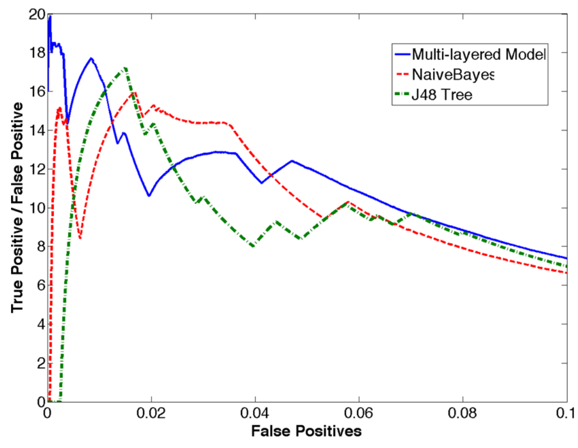

Fig. 12. ROC Curves Slope of Friendship Ranking

communications [14]. Similar to our resutls, they found that the gender had weak correlation with user communications. On the other hand, they found that the age attribute had non-trivial positive correlation, which is not the case in the Brightkite network. The difference may be due to that instant messaging network is a one-to-one more intimate media while Brightkite provides a public micro-blogging broadcasting media.

C. Top Recommendations

For friend recommendation application, however, we are more interested in how well to rank the user pairs at the **high ranking value** part. Because when we show a recommended friend list to a given user, she probably will only browse the top, say 100, suggestions and ignore the other ones. Considering the 18,951 active users, the top 10% or even top 1% recommended users are more important.

To show the performance of the three algorithms in high ranking value part, we changed the ROC curve slightly. We kept the x-axis as same as the ROC curve and changed the y-axis as true positives divided by false positives, which also can be considered as the slope of each point in ROC curve on the original point.

Figure 12 shows the result. We are interested in the range of false positive $\in (0, 0.1)$. The multi-layered model provides better performance than the other ones in most cases, in the range of false positives less than 0.01 and larger than 0.05. Especially, at the very top ranking set, the multi-layered model shows outstanding performance than the other two algorithms. The spikes in Figure 12 are the artifacts of sorting the user pairs of the same ranking values (corresponding to the x and y step increases in Figure 11).

Table IV shows the detailed results of the top friend pairs predictions. For example, in the third row of Table IV, the predicted number of friend pairs using the multi-layered model is 95 and there are 100 non-friend pairs, which means that in those top 195 ranked user pairs, there are 95 true friend pairs thus the true friend pairs percentage is 48.7%. At the same time, the Naive Bayes only provided 33.3% true friend pairs.

NF[a]	Friend Pairs Number and Percentage					
	Multi-layered		NaiveBayes		J48 Tree	
10,000	3,768	27.4%	3,380	25.3%	3,554	26.2%
1,000	857	46.1%	656	39.6%	790	44.1%
100	95	48.7%	50	33.3%	0	0
50	51	50.5%	0	0	0	0

TABLE IV
PERFORMANCE COMPARISONS

[a]Number of non-friend pairs.

For the J48 Tree, there was no true friend pairs in the top 100 ranked pairs. In the forth row of Table IV, multi-layered model successfully predicted 51 true friend pairs in the top 100 ranked pairs. On the other hand, neither Naive Bayes nor J48 Tree algorithm predicted any true friend pairs.

VII. Discussions

Unlike what other OSN studies suggested [11], [14], [19], the Brightkite social graph at this stage is still quite disconnected. There might be several contributing reasons. Brightkite is a young startup and its service has not attracted as many users as other popular OSNs. For example, our analysis of two-month Brightkite data (August and September 2008) shows that 41% "try-and-leave" users sent only one or two updates and then became inactive [15]. Some users may also simply use Brightkite as a location check-in service that feeds into their existing OSNs, such as Facebook and Twitter, without building their connections on Brightkite. We believe that the social graph will become more connected as the user base increases with location-based MSNs getting more popular. As the service evolves, some parameters in our model may become less effective and the model may need to be reconstructed using new data. However, we do not expect this model rebuilding to be frequent as the evaluation results suggest that the model worked well after 45 days.

In the proposed multi-layered model, we assume that the three metrics are independent, which is not necessarily true in reality due to natural correlation of the human dynamics. While the correlation does not reduce the model performance, other models that can exploit the inherent metric correlation, such as some form of neural network analysis, may further improve the results. Additionally, there may be other metrics that could be helpful for friendship modeling. For example, in Section V-C, we intuitively selected the location metric shown as Equation 18. An alternative approach may be to separate the geographical map into different regions based on population density and count the user pairs updates that belong to the same region. To experimentally find better modeling metrics remains as our future work.

Based on the proposed multi-layered model, we can design and implement a friend recommendation application to help users make new friends on Brightkite. One of the challenges is how to design the recommend application to support near real-time user requests. When a user logs in the Web site and clicks the given link, the current recommended friend list should be shown quickly. Here we discuss the possibility of implementing the recommendation system without using an expensive super computer or a computer cluster.

Since the tags in users' profile usually represent the users' interests, hobbies, and job, we assume that few users change that information frequently. We can compute each user pairs' tag metrics off-line and save the results into a database for querying. After the initial calculation, this information does not need to be recalculated unless new users are added or a user changes her tags.

Similar to the tag metric, the social graph metric of each user pair can also be calculated off-line at the beginning. After that, we only need to compute the updates incrementally. Since we calculated the rank factor of social graph F_s when their distance is 2 or 3, and ignored the other situations (Section V-A), we can prune our calculation into a small set when friend pairs change. In current Brightkite social graph, the users' average friend number is 6.38 and the standard deviation is 16.69. Thus the calculation overhead for changing friend pairs is not a concern.

It is harder, however, to calculate the location metrics of daily active users (averagely 3,656) with all active users (18,951). In fact, during the data collection period, (Section IV), we found 41,014 users, which may continue to increase as the popularity of the service increases. The amount of daily updated pairs is about 7×10^7. Instead of calculating each possible pairs, we can categorize the users based on the updates' location, and update the user pairs who are in the same category. The categorization can prune the calculation space because in our model, we only calculate the location rank factor when uses' updates are less than 30 miles.

Besides the public mode, in Brightkite, users can also send updates in protected mode, which is the default mode for new users. In this mode, all user's updates can only be seen by her friends. Our collected data set did not include those protected updates. Even if the protected updates may show different characteristics from the public updates, we can still apply the same methodology to build the multi-layered friendship model.

VIII. Conclusion and Future Work

This paper presents a multi-layered friendship model for location-based MSNs. We used real-world MSN data to build and evaluate the proposed model. By comparing with classic data mining algorithms, we found that the multi-layered model provided better performance, especially in top rank predictions, which is practical and valuable for a friend recommendation application.

Our future work will focus on implementing a friend recommendation application based on the proposed multi-layered model. Using the real system, we will be able to receive feedback from the users directly about the accuracy of this model. In addition, by combining other user attributes, such as those based on update content, we may further improve the modeling performance.

Acknowledgments

We thank our shepherd Sharad Agarwal and the anonymous reviewers for useful comments that improved this paper.

References

[1] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(47), 2002.

[2] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 43–52, 1998.

[3] G. Chen and F. Rahman. Analyzing privacy designs of mobile social networking applications. In *Proceedings of the IEEE/IFIP International Symposium on Trust, Security and Privacy for Pervasive Applications (TSP)*, Shanghai, China, Dec. 2008.

[4] H. Chun, H. Kwak, Y.-H. Eom, Y.-Y. Ahn, S. Moon, and H. Jeong. Comparison of online social relations in volume vs interaction: A case study of Cyworld. In *Proceedings of the 8th ACM Conference on Internet Measurement*, pages 57–70, Vouliagmeni, Greece, 2008.

[5] S. Gaonkar, J. Li, R. R. Choudhury, L. Cox, and A. Schmidt. Microblog: Sharing and querying content through mobile phones and social participation. In *Proceedings of the ACM MobiSys*, pages 174–186, Breckenridge, CO, June 2008.

[6] S. Gregory. An algorithm to find overlapping community structure in networks. In *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, pages 91–102, Sept. 2007.

[7] R. Grob, M. Kuhn, R. Wattenhofer, and M. Wirz. Cluestr: Mobile social networking for enhanced group communication. In *Proceedings of the International Conference on Supporting Group Work*, Sanibel Island, FL, May 2009.

[8] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004.

[9] A. Hirsch. iPhone 2.0 apps: The social networking app comparison. Mashable: The Social Media Guide, July 2008.

[10] L. Humphreys. Mobile social networks and social practice: A case study of Dodgeball. *Journal of Computer-Mediated Communication*, 13(1):341–360, October 2007.

[11] A. Java, X. Song, T. Finin, and B. Tseng. Why we Twitter: Understanding microblogging usage and communities. In *Proceedings of the First Workshop on Web Mining and Social Network Analysis*, Aug. 2007.

[12] G. H. John and P. Langley. Estimating continuous distributions in bayesian classifiers. In *In Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 338–345. Morgan Kaufmann, 1995.

[13] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins. Microscopic evolution of social networks. In *Proceeding of the ACM KDD*, Las Vegas, NV, Aug. 2008.

[14] J. Leskovec and E. Horvitz. Worldwide buzz: Planetary-scale views on an instant-messaging network. Technical Report MSR-TR-2006-186, Microsoft Research, June 2007.

[15] N. Li and G. Chen. Analysis of a location-based social network. Submitted, Apr. 2009.

[16] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76–80, Jan/Feb 2003.

[17] H. Liu, P. Maes, and G. Davenport. Unraveling the taste fabric of social networks. *International Journal on Semantic Web and Information Systems*, 2(1):42–71, 2006.

[18] Location-based social networking to generate $3.3 billion by 2013. ABI Research Market Report, Aug. 2008.

[19] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM Conference on Internet Measurement*, pages 29–42, New York, NY, USA, 2007. ACM.

[20] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[21] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, pages 175–186. ACM Press, 1994.

[22] M. F. Schwartz and D. C. M. Wood. Discovering shared interests using graph analysis. *Communications of the ACM*, 36(8):78–89, Aug. 1993.

[23] Social networking and commerce draw consumers into the mobile Web. M:Metrics Market Report, May 2008.

Digital Object Identifier: 10.4108/ICST.MOBIQUITOUS2009.6828
http://dx.doi.org/10.4108/ICST.MOBIQUITOUS2009.6828